

Tentative de formalisation algorithmique de la démarche du phonologue

Un outil d'aide à la formulation d'hypothèses phonologiques

Michel Jacobson

LACITO/CNRS

8, rue Guy Moquet – 94800 Villejuif

Tél.: ++33 (0)1 49 58 27 21 - Fax: ++33 (0) 1 49 58 37 79

Mél: jacobson@idf.ext.jussieu.fr - <http://lacito.vjf.cnrs.fr>

ABSTRACT

We present a formal computerized model of a particular linguistic theory, functional phonology -- a theory which is often criticized precisely for its lack of formalization. This theory proposes on the one hand a general framework for the expression of phonological phenomena and on the other a model for a discovery procedure for phonological units. In formalizing this theory explicitly, we have arrived at (1) a formalism for the expression of data and hypotheses and (2) a computer program emulating the functionalist methods of phonological analysis.

In the paper, we present the principal data structures used and the procedures which we have designed to process them. Methodological obstacles which we have faced in implementing the model are discussed.

RÉSUMÉ

Nous présentons ici la formalisation informatique d'une théorie linguistique particulière (le fonctionnalisme) souvent critiquée justement par son absence de formalisation. Cette théorie propose non seulement un cadre général pour exprimer les phénomènes d'ordre phonologique mais aussi une modélisation de la démarche de découverte des unités linguistiques. Le travail d'explicitation de cette théorie a débouché 1) sur la mise en place d'un formalisme d'expression pour les données et les hypothèses 2) sur la conception d'un logiciel reproduisant ou assistant la démarche de l'analyse phonologique de cette école.

Nous présentons dans ce papier les principales structures de données ainsi que les principales fonctions du logiciel qui y font appel. Nous essaierons enfin de relever les différents obstacles méthodologiques que nous avons rencontrés dans notre démarche de formalisation.

1. PRÉSENTATION

Le linguiste phonologue quand il débute l'analyse d'une langue qu'il ne connaît pas, ne dispose que de ses connaissances générales en linguistique ou en phonétique. Il connaît un certain nombre d'indices phoniques qu'il sait pouvoir mesurer ou estimer. Il sait aussi que certains de ces indices ont participé à la

définition de traits distinctifs dans des langues déjà décrites. Ce cadre théorique minimal peut se formaliser par une grille d'observation que le phonologue va utiliser pour décrire de nouvelles langues.

L'alphabet phonétique international (A.P.I.) a été une tentative de normalisation typographique d'une telle grille, permettant la transcription des langues orales. Des équivalents ont été proposés en informatique tel que SAMPA [Gib97], supplanté récemment par l'Unicode [Uni00]. Mais ces propositions ne répondent qu'à une partie du problème, celui du codage informatique des conventions typographiques. Aucun des deux n'offre la possibilité d'explicitier la structure du système de notation, c'est-à-dire de définir les dimensions phonologiques qui sont représentées typographiquement par l'organisation des unités, en lignes colonnes et tableaux dans l'A.P.I.. Nous verrons plus loin la proposition de formalisation que nous faisons de cette structure.

Les théories linguistiques sont relativement neutres quant au système de notation employé. Globalement, il y a un accord entre elles pour utiliser des systèmes de type alphabétique (A.P.I. ou autres). Les caractères représentent alors des regroupements d'événements et d'états stables caractéristiques. Elles diffèrent plus largement par l'analyse qu'elles font de ces segments alphabétiques. Ainsi les structuralistes, les fonctionnalistes et une partie des générativistes décomposent les segments en matrices de traits (binaires ou non), en inventaires (à caractère universel ou non). L'approche adoptée par la théorie du « charme et du gouvernement » [Kay88] décompose les segments en combinaisons d'éléments (matrices de traits entièrement spécifiées). Enfin les théories auto-segmentales vont, elles, analyser chaque segment sur plusieurs lignes à la fois (tires) représentant des dimensions d'analyses, etc. Nous avons donc besoin dans une formalisation informatique d'explicitier cette ré-analyse spécifique des segments notationnels en primitives de la théorie. Cette ré-analyse étant propre à chaque théorie, il n'était pas envisageable de trouver un formalisme qui convienne à toutes. Nous avons donc choisi de travailler uniquement dans le cadre d'une théorie particulière.

Dans le travail que nous présentons ici, nous allons examiner les concepts et la méthode de la phonologie fonctionnaliste [Mar56] dans la formalisation qu'en a

proposé Jean-Pierre Goudaillier [Gou86]. Celle-ci bien que déjà ancienne n'a donné lieu qu'à de rares tentatives d'automatisation. Nous proposons dans ce qui suit d'examiner cette théorie par le biais de la création conjointe d'un formalisme d'expression et d'un logiciel qui implémente la démarche des phonologues de cette école sous forme d'algorithmes .

2. MODÉLISATION INFORMATIQUE

La modélisation informatique d'une théorie linguistique passe par l'explicitation de toutes les primitives et de tous les concepts qu'elle manipule. La modélisation que nous proposons des procédures de découverte des unités, tout comme celle de mise en lumière des phénomènes phonologiques (variations combinatoires, neutralisations,...), est explicitée dans notre travail sous forme d'algorithmes, puis a été implémenté avec un langage de programmation pour former le logiciel décrit ici.

La structure des données est, consignée dans une syntaxe formelle qui définit à la fois les données d'entrées (le corpus), toutes les hypothèses (y compris celles définissant le système de notation utilisé pour le corpus) ainsi que les résultats.

2.1. Le codage des données

Pour être utilisable par notre outil, un corpus doit être constitué d'une liste d'items lexicaux avec pour chacun d'eux, au minimum, sa description phonétique, un identifiant de son sens et éventuellement sa position dans un fichier d'enregistrement. Cette description peut s'étendre bien sûr à de nombreux autres champs tels que ceux que l'on pourrait s'attendre à trouver dans un dictionnaire digne de ce nom (partie du discours, étymologie, etc.), mais seuls les premiers cités sont traités effectivement dans le programme.

```
<SIGNE id="w123">
  <SIGNIFIANT>pakt</SIGNIFIANT>
  <SIGNIFIE>pacte</SIGNIFIE>
  <AUDIO start="34.32" end="35.10">
</TRAIT>
```

Figure1 :Exemple d'entrée minimale d'un corpus.

Nous situons la constitution d'un tel corpus dans une étape phonétique ou pré-phonologique. Il s'agit de mesurer au cours du temps, dans un processus d'échantillonnage, les différents événements phonétiques ou indices.

2.2. Le codage des hypothèses

La structure du système de notation qui constitue notre grille d'analyse est bien sûr la première de nos hypothèses. Elle doit être explicitée en indiquant toutes les relations qui existent entre les segments et les objets manipulés par la théorie. Par exemple : Chaque trait est déclaré 1) soit en listant tous les segments qui sont

caractérisés par lui

```
<TRAIT name="occlusif">
  <INVENTAIRE>p t k b d g
...</INVENTAIRE>
</TRAIT>
```

Figure2 :Exemple de déclaration du trait 'occlusif'.

2) soit en listant les contraintes sous forme de conditions sur les traits déjà déterminés.

```
<TRAIT name="affriqué">
  <CONDITION>(occlusif)^(fricatif)</CONDI
TION>
</TRAIT>
```

Figure3 :Exemple de déclaration du trait 'affriqué'.

Les relations entre les traits sont elles aussi notées sous l'appellation de dimensions oppositionnelles spécifiant le type de relations qui les lient.

```
<DIMENSION name="degré d'aperture" ordre="true">
  <TRAITS>ouvert semi-ouvert...
fermé</TRAITS>
</DIMENSION>
```

Figure4 :Exemple de déclaration d'une dimension ordonnée.

L'approche fonctionnaliste utilise la notion de syllabe, notamment dans la description des contextes de pertinence, mais sans la définir comme un objet de la théorie. Or il est impossible dans le cadre que l'on s'est fixé d'utiliser une notion sans la définir de manière explicite. Comme cette notion est fréquemment utilisée par les phonologues de cette école nous avons choisi d'emprunter sa définition à d'autres approches théoriques. Dans notre travail, elle est exprimée, avec les autres hypothèses, sous la forme de contraintes sur l'enchaînement des types de segments. Ces contraintes peuvent être synthétisées par un automate à états finis (ou canon syllabique).

```
<SYLLABE type="canon">P?[SL]?G?V[PSLG]*</SYLLABE>
```

Figure5 :Canon syllabique utilisé pour le français.

(Les lettres 'PSLGV' représentent respectivement les classes : Plosive, Sonante, Liquide, Glyde etVoyelle)

Une autre solution implémentée dans notre outil consiste à définir un ordre des structures préférentielles comme on pourrait le faire dans le cadre de la théorie de l'optimalité [Pri93].

Une dernière hypothèse importante est la définition des contextes de pertinence. Ceux-ci sont eux exprimés comme précédemment sous forme de contraintes (clauses logiques exprimées sous forme normale conjonctive) qui portent à la fois sur la position des segments entre eux ou au sein de la syllabe et sur la composition en termes de traits de ces derniers.

```
<CONTEXTE nom="voyelles en syllabe fermée">
  <contrainte pos="0" type="seg" >(voyelle)^(–nasalise)...
  <contrainte pos="+1" type="syll" >(./#)...
</CONTEXTE>
```

Figure6 :Exemple de définition du contexte

« Voyelles orales en syllabe fermée »

2.3. Les traitements

Le logiciel décompose la démarche du phonologue fonctionnaliste en un certain nombre d'étapes successives qui sont parfois répétées de manière cyclique.

- Les deux premières étapes sont la segmentation et la permutation. Elles vont opérer des regroupements de segments en fonction respectivement du système de notation et de la combinatoire utilisée dans la langue. L'enchaînement de ces deux étapes sera opéré en boucle jusqu'à ce que toutes les unités minimales obtenues soient permutable ou jugées comme telles.

Figure7:Enchaînement des opérations de permutation et de commutation.

Si un segment ne commute pas, c'est-à-dire si l'on peut observer une influence entre ce segment et son environnement proche, il est alors concaténé avec ses plus proches voisins puis le corpus est à nouveau segmenté et les nouveaux regroupements testés par l'opération de permutation. Par exemple le segment [Σ] de l'espagnol n'admet pas d'autres segment à sa gauche que le segment [t]. Le segment complexe [tΣ], lui en revanche, satisfait le test de permutation. Cette optique, adoptée dans une première version du logiciel, a été abandonnée au profit d'une démarche assistée par le linguiste. En effet, en général, ni le corpus, ni même la langue ne sont suffisants pour que tous les éléments permurent. Avec notre outil, le linguiste opère l'opération de permutation en définissant un niveau de contrainte (combien de types de permutation et éventuellement lesquels doivent être satisfaits). Les résultats de la permutation ne sont délivrés qu'à titre d'information et le linguiste doit en déduire quels sont les regroupements de segments qu'il doit opérer.

- L'algorithme de commutation permet d'examiner les influences des unités minimales entre elles en observant leurs comportements à l'intérieur des contextes où elles s'expriment. Dans un premier temps les unités issues de l'étape précédente sont ventilées dans des inventaires contextuels en fonction de leurs conditions d'apparition dans le corpus. La définition des contextes est une tâche qui relève des connaissances du linguiste et qu'il exprime sous forme d'hypothèses (Cf. Figure 6). Une fois ces inventaires établis, ils forment des paradigmes dans lesquels on va chercher à valider le plus d'oppositions possibles entre membres. C'est la présence ou non de « paires minimales » dans le corpus qui va permettre cette validation. Si la commutation est possible entre deux unités c'est qu'il s'agit de deux phonèmes différents et les traits qui les identifient l'un par rapport à l'autre ont alors une fonction distinctive. Par exemple en français l'opposition « choux » ~ « joue » (Cf. Figure 8) permet de définir les 2 candidats phonèmes (/(\Sigma)/ et /(Z)/) ainsi

que de valider les traits 'sourd' et 'sonore' comme distinctifs dans le contexte « initiale ». Cette étape a pour objectif de définir en termes de traits distinctifs la place que les unités occupent au sein des sous-systèmes contextuels.

- Le dernier algorithme est celui du rapprochement des inventaires. Celui-ci opère sur la base de la définition des unités issues de la commutation en termes de traits distinctifs. C'est au cours de cette étape que l'on pourra distinguer les cas de variantes combinatoires, les cas de neutralisation, les assimilations, les distributions lacunaires, etc. Par exemple, dans le français du midi (variété toulousaine) les voyelles moyennes [e] et [E] qui apparaissent respectivement dans les contextes « syllabe fermée » et « syllabe ouverte » partagent la même définition : 'étirée' 'avant' '2^{ème} degrés/3'. L'outil en déduit donc qu'il s'agit d'un seul et même phonème dont les réalisations contextuelles diffèrent.

3. IMPLÉMENTATION LOGICIELLE

Le logiciel mis au point pour implémenter l'analyse de la méthode fonctionnaliste est écrit en Java. Il ne sait traiter qu'un type particulier de fichier XML [Bra98], ceux qui respectent la syntaxe (ou DTD : Document Type Definition) définie à cet effet. Quelques facilités ont été ajoutées au programme pour consulter directement les enregistrements audio afin d'aider à la confrontation de l'analyse et des données.

Figure8:Image du logiciel (paires minimales opposant [Σ] et [Z] du français en contexte initial absolu).

4. BILAN

Le formalisme choisi pour coder le corpus, les hypothèses et les résultats nous permet, entre autres avantages, 1) de structurer explicitement ces derniers 2) de noter n'importe quel caractère existant dans Unicode (potentiellement plus de 65000 caractères avec notamment tous ceux de l'API), 3) d'être indépendant de la plate-forme, 4) de bénéficier de tous les développements faits dans ce domaine (parsers, éditeurs, langages de requêtes, etc.).

Le but de cet outil est qu'il reproduise la méthode fonctionnaliste telle que nous avons pu la modéliser. Il est par exemple capable de lister tous les phonèmes d'une langue par simple application des hypothèses sur un corpus et ceci sur la base des principes structuraux (cases vides, corrélations, etc.) et des principes fonctionnels (économie) de la théorie fonctionnaliste. Il a été conçu comme un outil de validation d'hypothèses, qui permet de faire des aller-retours entre des hypothèses et leurs validations dans le but d'ajuster et d'affiner ces hypothèses.

Une autre préoccupation dans la création de cet outil a été l'automatisation d'un certain nombre de tâches très répétitives et fastidieuses pour le linguiste, telles que le comptage des fréquences d'apparition de certains

segments dans un contexte particulier ou bien la recherche de paires minimales.

Enfin, cet outil se veut ouvert vers d'autres utilisations, telle que la confrontation des hypothèses avec les données brutes d'enregistrement. Il est à tout moment possible d'écouter un mot ou bien d'exporter le signal vers l'outil d'analyse phonétique de son choix. Il est aussi possible à moindre coût (c'est-à-dire au moyen de formatages ou de transformations XSL-T) d'exporter ou d'importer des ressources (dictionnaires ou lexiques) qui utilisent d'autres types de codage explicites (SGML, XML, Lexware, Shoebox, etc.)

LIMITES ET PERSPECTIVES

Nous avons rencontré dans notre démarche de formalisation informatique un certain nombre d'obstacles méthodologiques. Nous pouvons citer parmi eux notamment l'absence de définition de la syllabe, une conception unidimensionnelle des traits et de leurs relations (par opposition à la théorie de la géométrie des traits). Enfin, la phonologie fonctionnaliste est une phonologie du segment, unité postulée, alors que la véritable unité utilisée est le trait, ce qui rend difficile toute analyse dépassant le cadre du segment comme les assimilations à distance (par exemple : les harmonies vocaliques).

Un des avantages de notre approche est la rigueur que l'informatique nous a obligé à apporter à notre formalisation. Cette même démarche pourrait être adoptée pour implémenter des modélisations issues d'autres théories phonologiques. Mais celles-ci proposent rarement une description du mécanisme de découverte des unités elles mêmes. En général, elles proposent uniquement un cadre conceptuel dans lequel on va pouvoir exprimer de différentes manières (sous forme de règles, de contraintes, de hiérarchies de contraintes, etc.) les mécanismes phonologiques mis en œuvre dans les langues. Il serait possible dans de tels cas d'aménager le programme pour que les hypothèses sur les mécanismes phonologiques soient directement renseignées par le linguiste. La fonction principale du logiciel serait alors uniquement d'effectuer le passage d'une écriture de type phonétique à une écriture de type phonologique et vice versa, ce qui est une fonction qu'il remplit déjà mais une fois les unités linguistiques déduites du corpus.

C'est dans cette optique que nous avons commencé à travailler avec la théorie de l'optimalité pour la modélisation de la notion de syllabe. Ce travail est d'autant plus aisé que cette théorie bénéficie entre autre avantage d'une formalisation assez poussée.

BIBLIOGRAPHIE

[Bra98] Bray T., Paoli J., Sperberg-McQueen C.M. (Eds.), 1998. *Extensible Markup Language (XML) 1.0*. W3C Recommendation, 10

février 1998
<http://www.w3.org/TR/REC-xml>.

- [Gib97] Gibbon D., Moore R. et Winski R. (Eds). *Handbook of Standards and Resources for Spoken Language Systems*. Berlin: Mouton de Gruyter, 1997.
- [Gou86] Goudaillier J-P. Eléments de phonologie fonctionnelle expérimentale. *Langues et Linguistique*, 12. Québec: Université Laval, 1986, pp 131-180.
- [Kay88] Kaye J., Lowenstamm J. et Vergnaud J-R. La structure interne des éléments phonologiques: une théorie du charme et du gouvernement. *Recherches Linguistiques*, 17. 1988, p 109-135.
- [Mar56] Martinet A. *La description phonologique avec application au franco-provençal d'Hauteville (Savoie)*. Genève: Droz, 1956.
- [Pri93] Prince A. et Smolensky P. *Optimality Theory: Constraint Interaction in Generative Grammar*. RuCCS Technical Report 2. Piscataway, NJ: Rutgers University, and Boulder, CO: Department of Computer Science, University of Colorado 1993.

[Uni00]